数理解析研究所講究録
第 1829 巻 2013 年 156-162

Online TSP for a Class of Pseudo-Planar Graphs

Yuya Higashikawa[1], Naoki Katoh[1*], and Seok-Hee Hong[2**]

[1] Department of Architecture and Architectural Engineering, Kyoto University, Japan, {as.higashikawa,naoki}@archi.kyoto-u.ac.jp,

[2] School of Information Technologies, University of Sydney, Australia, shhong@it.usyd.edu.au

Abstract. This paper considers online TSP in a pseudo-planar graph, say a maximal 1-plane geometric graphs. A maximal 1-plane geometric graph is a geometric graph such that each edge of the graph crosses the other edge at most once and any graph obtained by adding a new edge to the graph is no more 1-plane graph. Suppose that a searcher is required to visit all vertices of the given graph. He/she starts the exploration from a given vertex and finally returns to the initial vertex as quickly as possible. The information of the graph is given online. As the exploration proceeds, a searcher gains more information of the graph. We give a competitive analysis of algorithms in [2], [3] for a maximal 1-plane geometric graph, and we prove an upper bound of a competitive ratio as 16.

Keywords: online algorithm, traveling salesman problem, competitive analysis, 1-planar graph, maximal 1-planar graph, maximal 1-plane geometric graph

1 Introduction

We study *online traveling salesman problems* (**online TSP** for short) for a maximal 1-plane geometric graph.

Online TSP in an undirected graph are defined as follows. Given an undirected graph $G = (V, E)$, suppose that a searcher is initially at a vertex of G. Starting from the origin $o \in V$, the aim of a searcher is to visit all vertices of G at least once and to return to o as quickly as possible. A searcher makes all his/her decisions based on partial knowledge obtained so far with respect to the graph and gathers new information as exploration proceeds. We assume that vertices are labeled so that a searcher can distinguish them. The length of an edge $e \in E$ is denoted by $|e|$. We also assume the ability of a searcher as follows: whenever a searcher visits a new vertex, he/she learns all incident edges, their lengths and the labels of their end vertices. The goal is to find a tour of minimum length that visits all vertices and returns to the origin.

In this paper, we consider exploring a maximal 1-plane geometric graph. For a undirected graph $G = (V, E)$ embedded on the plane, G is called a *geometric graph* if each edge of G is drawn as a straight line segment connecting two end vertices of the edge. For a undirected graph $G = (V, E)$, G is called a *k-planar graph* if it can be drawn on the plane such that each edge of G is crossed by other edges at most k times. Also for an undirected graph $G = (V, E)$ embedded on the plane, G is called a *k-plane graph* if each edge of G is crossed by other edges at most k times. In the following, for a k-plane graph $G = (V, E)$, an edge of G is said to be a *blue edge* if it crosses another edge, and to be a *red edge* otherwise. Then there are two definitions of the maximality

* Supported by JSPS Grant-in-Aid for Scientific Research(B)(21300003)
** Supported by ARC DP0881706 and ARC DP0988838

of k-plane graphs. In general definition (Suzuki [4]), for a k-plane graph $G = (V, E)$, G is called a *maximal k-plane graph* if adding any new edge to G produces an edge with at least $k + 1$ crossing. In the other definition (Eades et al. [1]), for a k-plane graph $G = (V, E)$, G is called a *red-maximal k-plane graph* if any red edge cannot be added to G. This paper adopts the former definition. Furthermore we restrict a graph class to that of geometric graphs. For a geometric graph, the k-planarity and the maximal k-planarity can be similarly defined. Namely, for a geometric graph $G = (V, E)$, G is called a *k-plane geometric graph* if G is a k-plane graph, and G is called a *maximal k-plane geometric graph* if G is a maximal k-plane graph. For example, an embedded graph in Fig. 1 is a maximal 1-plane geometric graph, however it is a planar graph (see Fig. 2). In general, the performance of an online algorithm is measured by a competitive

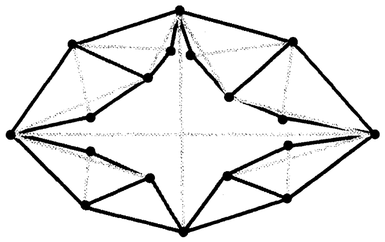

Fig. 1. A maximal 1-plane geometric graph (dark grey edges represent red edges while light grey edges represent blue edges)

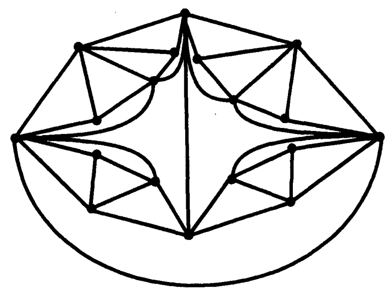

Fig. 2. A planar graph

ratio which is defined as follows. Let $\mathcal{S}$ denote a class of objects to be explored. When an online exploration algorithm ALG is used to explore an object $S \in \mathcal{S}$, let $|\mathsf{ALG}(S)|$ denote the tour length (cost) required to explore S by ALG. Let $|\mathsf{OPT}(S)|$ denote the tour length (cost) required to explore S by the offline optimal algorithm. Then the competitive ratio of ALG is defined as follows:

$$\sup_{S \in \mathcal{S}} \frac{|\mathsf{ALG}(S)|}{|\mathsf{OPT}(S)|}.$$

For online TSP in an undirected graph, Kalyanasundaram et al. [2] presented an algorithm $\mathsf{ShortCut}$. They showed that this algorithm achieves 16-competitive for an undirected planar graph. Recently, Megow et al. [3] sophisticated the formulation of $\mathsf{ShortCut}$ and made the competitive analysis simple. They called their formulation of $\mathsf{ShortCut}$ newly $\mathsf{Blocking}_\delta$. Also they generalized the result in [2] to $16(1 + 2g)$-competitive for an undirected graph with genus g.

We give a competitive analysis of $\mathsf{Blocking}_\delta$ algorithm in [3] for online TSP in a maximal 1-plane geometric graph. In [3], for a set of edges P which $\mathsf{Blocking}_\delta$ traverses and a minimum spanning tree MST of the entire graph, they showed that a competitive ratio of their algorithm is at most 16 if $P \cup MST$ is planar. We show that $P \cup MST$ is also planar for a maximal 1-plane geometric graph and hence that 16-competitiveness follows for this class of non-planar graphs. Upper bound of genus of a maximal 1-plane geometric graph is non-trivial and has not been known yet, thus we cannot apply directly the result of [3] to our case. However, even if genus of a maximal 1-plane geometric graph is only 1, we improve a competitive ratio for such a graph from 48 to 16.

2 Blocking$_\delta$ algorithm

In this section, we briefly review the graph exploration algorithms of [2] and [3]. Although the algorithm of [3] is essentially the same as that of [2], we will review the one by [3] because it sophisticated the one by [2]. The following description is based on [3].

Definition 1 *A vertex is said to be* explored *if it has been visited at least once by a searcher, and* unexplored *otherwise. An edge is said to be* explored *if both end vertices are explored. A* boundary *edge uv is an edge with an explored end vertex u and an unexplored end vertex v.*

Definition 2 *For a fixed parameter $\delta > 0$, a boundary edge $e = uv$ is said to be* blocked *if there is a boundary edge $e' = u'v'$ with u' explored and v' unexplored such that $|e'| < |e|$ holds and the length of any shortest known path from u to v' is at most $(1+\delta)|e|$.*

The algorithm of [3] is named as Blocking$_\delta$. It can be seen as a sophisticated variant of depth-first-search (DFS for short). The crucial ingredient is a blocking condition depending on a fixed parameter $\delta > 0$, which determines when to diverge from DFS. The procedure of Blocking$_\delta$ for a partially explored graph G and a vertex y of G which is explored for the first time, say Blocking$_\delta(G, y)$, is represented as follows.

Algorithm 1 The exploration algorithm Blocking$_\delta(G, y)$ (by [3])

Input: A partially explored graph G and a vertex y of G which is explored for the first time.
while there is an unblocked boundary edge $e = uv$, with u explored and v unexplored, such that $u = y$ or such that e had previously been blocked by some edge xy **do**
  walk a shortest known path from y to u
  traverse $e = uv$
  Blocking$_\delta(G, v)$
  walk a shortest known path from v to y
end while

Blocking$_\delta$ performs a standard DFS, but it traverses a boundary edge only if it is not blocked. Suppose that a searcher is at a vertex u and considers traversing a boundary edge uv. If uv is blocked, then its traversal is postponed, possibly forever; otherwise a searcher traverses uv. Traversing xy and exploring y may cause another edge uv, whose traversal was delayed earlier, to become unblocked. Then a searcher walks a shortest known path from y to u and traverses $e = uv$. To explore the entire graph starting from the origin o, we call Algorithm 1 as Blocking$_\delta(G_o, o)$, where G_o is the partially explored graph in which only o has been visited so far.

Theorem 1 (by [3]) *A competitive ratio of* Blocking$_2$ *for an undirected planar graph is at most* 16.

Sketch of proof in [3]. Let P denote a set of edges which $\mathsf{Blocking}_\delta$ traverses at line 3 for each iteration of the while loop. Actually a searcher may traverse edges at lines 2, 3 and 5. Suppose that at line 1 uv had previously been blocked by some edge xy, then the length of a path which a searcher moves at line 2 is at most $(1+\delta)|e|$ from Definition 2. Thus the total length of edges which he/she traverses at line 2 and 3 is at most $(2+\delta)|e|$. Considering that at line 5 he/she can traverse backward same edges as at lines 2 and 3, the length of edges traversed in each iteration of the while loop is at most $2(2+\delta)|e|$. Therefore the tour length required to explore an undirected planar graph G by $\mathsf{Blocking}_\delta$, say $|\mathsf{Blocking}_\delta(G)|$, satisfies the following inequality:

$$|\mathsf{Blocking}_\delta(G)| \leq 2(2+\delta)|P|. \tag{1}$$

Let MST be a minimum spanning tree that shares a maximum number of edges with P. Then considering that $P \cup MST$ is planar and so each edge $e \in P \setminus MST$ is contained in at most two face cycles, for each edge $e \in P \setminus MST$ one of its face cycles can be uniquely assigned as C_e such that every assigned cycle is different from each other. By [3], the following claim is proved.

Claim 1 (by [3]) *If an edge $e \in P \setminus MST$ is contained in a cycle C in $P \cup MST$, then the cycle C has length at least $(2+\delta)|e|$.*

From this claim, $(2+\delta)|P \setminus MST| \leq \sum_{e \in P \setminus MST} |C_e|$ holds, and also $\sum_{e \in P \setminus MST} |C_e| \leq 2|P \cup MST| = 2(|MST| + |P \setminus MST|)$ holds, thus we have $|P \setminus MST| \leq (2/\delta)|MST|$, namely,

$$|P| \leq (1 + \frac{2}{\delta})|MST|. \tag{2}$$

From (4) and (2), we obtain

$$|\mathsf{Blocking}_\delta(G)| \leq 2(2+\delta)(1+\frac{2}{\delta})|MST|. \tag{3}$$

Since the tour length required to explore G by the offline optimal algorithm, say $|\mathsf{OPT}(G)|$, satisfies $|\mathsf{OPT}(G)| \geq |MST|$ and $2(2+\delta)(1+2/\delta)$ is at least 16 for $\delta = 2$, we can see $\mathsf{Blocking}_2$ is 16-competitive for an undirected planar graph. □

3 Competitive analysis

Let $G = (V, E)$ be a maximal 1-plane geometric graph. For any two vertices $u, v \in V$, let uv denote a straight line segment between u and v. Notice that uv denotes an edge if u and v are adjacent with each other in G. For any connected subgraph $G' \subseteq G$, let $MST(G')$ denote a minimum spanning tree of G'. Then the following proposition holds.

Proposition 1 *For an undirected connected graph $G = (V, E)$ with weights associated with edges, consider a connected subgraph G' and $MST(G')$. If an edge e of G' does not belong to $MST(G')$, e does not belong to $MST(G)$, either.*

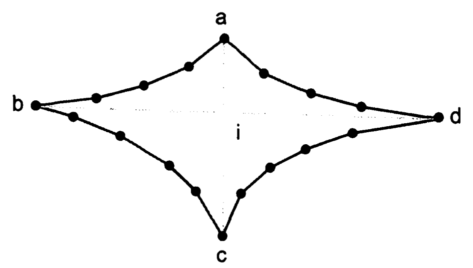

Fig. 3. A partial structure around a pair of blue edges

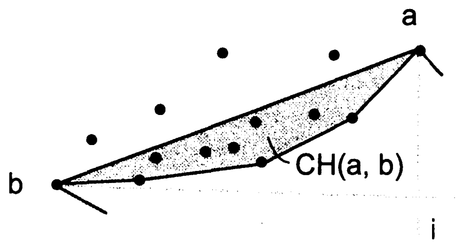

Fig. 4. $CH(a, b)$

At first, we consider a partial structure around a pair of blue edges ac and bd which intersect each other at a point i (see Fig. 3). For a triangle abi in Fig. 3, let S denote a set of vertices strictly lying in the inside of abi. For a vertex set $S \cup \{a, b\}$, let $CH(a, b)$ denote the convex hull for $S \cup \{a, b\}$ (see Fig. 4). If $S = \emptyset$, let $chain(a, b)$ denote an edge ab. If $S \neq \emptyset$, let $chain(a, b)$ denote the boundary path from a to b of $CH(a, b)$ which is different from the boundary path consisting of an edge ab. In both cases there is no edge which crosses $chain(a, b)$, so there are red edges along $chain(a, b)$ because of the maximality of G. We can similarly define $chain(b, c)$, $chain(c, d)$ and $chain(d, a)$. We have the following lemma.

Lemma 1 *For a pair of blue edges ac and bd, there exist always four concave chains of red edges (each chain may possibly consist of one red edge), $chain(a, b)$, $chain(b, c)$, $chain(c, d)$ and $chain(d, a)$ for short, such that all chains lie in the inside of a quadrilateral $abcd$ and no vertex exists in the inside of a polygon formed by these four concave chains.*

Let G^* denote a subgraph of G which consists of two blue edges, ac and bd, and four concave chains of red edges, $chain(a, b)$, $chain(b, c)$, $chain(c, d)$ and $chain(d, a)$. Assume without loss of generality that $|ai| = \min\{|ai|, |bi|, |ci|, |di|\}$ holds. Then we have the following lemmas.

Lemma 2 *A blue edge bd is not contained in $MST(G)$.*

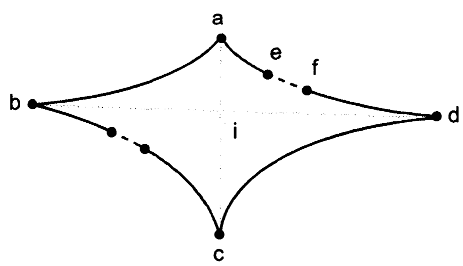

Fig. 5. Illustration of the subgraph G^* used in the proof of Lemma 2

Proof. Suppose otherwise. By the contraposition of Proposition 1, bd is also contained in $MST(G^*)$, so there is one red edge, say ef, which is on the path consisting of two concave chains, $chain(a,b)$ and $chain(d,a)$, and is not contained in $MST(G^*)$ (see Fig. 5). The length of $chain(a,b)$ is less than $|ai|+|bi|$, similarly the length of $chain(d,a)$ is less than $|ai|+|di|$, thus

$$|ef| < \max\{|ai|+|bi|, |ai|+|di|\} \tag{4}$$

holds. By (4) and the assumption of $|ai| \le |di|$ and $|ai| \le |bi|$, we have

$$|ef| < |bi|+|di| = |bd|. \tag{5}$$

From (5) $(MST(G^*) \setminus \{bd\}) \cup \{ef\}$ is another spanning tree of G^* whose length is less than that of $MST(G^*)$, which contradicts the minimality of $MST(G^*)$. □

Lemma 3 *For $\delta \ge 1$,* Blocking$_\delta$ *does not traverse a blue edge bd.*

Proof. Suppose that bd is a boundary edge such that b is explored and d is unexplored. Then there is one boundary edge, say ef, on the concave chain path from b via a to d such that all vertices on the concave chain path from b to e is explored (see Fig. 6). We show that bd is blocked by ef as follows. At first, we have $|ef| < |bd|$ from (5).

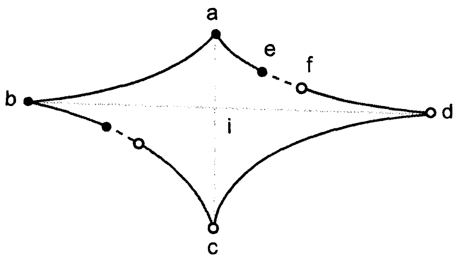

Fig. 6. Illustration of the case that bd is a boundary edge

Secondly, let $SP(b,e)$ denote the shortest known path from b to e, then we have the following inequality:

$$\begin{aligned}|SP(b,e)| &\le |ai|+|bi|+|ai|+|di| \\ &\le 2|bd|. \end{aligned} \tag{6}$$

From (6) and $\delta \ge 1$, we obtain $|SP(b,e)| \le (1+\delta)|bd|$. Therefore bd is always blocked by ef if bd is a boundary edge, so Blocking$_\delta$ does not traverse bd. □

Theorem 2 *A competitive ratio of* Blocking$_\delta$ *for a maximal 1-plane geometric graph is at most 16.*

Proof. As in [3], let P denote a set of edges which Blocking$_\delta$ traverses at line 3. Also in [3], they proved that a competitive ratio of Blocking$_\delta$ is at most 16 if $P \cup MST(G)$ is a planar graph. From Lemmas 2 and 3, we showed that at least one edge for each pair of blue edges is never included in P and in $MST(G)$. Thus we obtain $P \cup MST(G)$ is planar. □

4 Conclusion

We give a competitive analysis of algorithms in [2] and [3] for online TSP in a maximal 1-plane geometric graph, and we prove a competitive ratio is at most 16.

References

1. P. Eades, S. Hong, G. Liotta and S. Poon, "Straight-line Drawings of 1-planar Graphs", *Technical report IT-IVG-2011-01(School of Information Technologies, University of Sydney)*, 2011.
2. B. Kalyanasundaram and K. R. Pruhs, "Constructing competitive tours from local information", *Theoretical Computer Science*, 130, pp. 125-138, 1994.
3. N. Megow, K. Mehlhorn and P. Schweitzer, "Online graph exploration: New results on old and new algorithms", In *Proc. 38th ICALP* (LNCS 6756), pp. 478-489, 2011.
4. Y. Suzuki, "Re-embeddings of Maximum 1-Planar Graphs", *SIAM J. on Discrete Mathematics*, 24(4), pp. 1527-1540, 2010.